KaSim reference manuel

J. Krivine

November 3, 2010

Contents

Chapter 1

Introduction

KaSim is an open source stochastic simulator of rule-based models [3, 2, 4] written in the κ-calculus. Basically KaSim takes one or several kappa files as inputs and generates stochastic trajectories of various observable. KaSim implements the network free simulation algorithm for rule-based models [1] that extends Gillespie's algorithm [5, 6].

A *simulation event* corresponds to the application of a rewriting rule, contained in the kappa file, to the current graph (also called a *mixture*). The rule is selected according to its activity, *i.e* the number of instances it has in the current mixture, multiplied by its kinetic rate, and applied one of its possible instance in the graph. It results in a new graph together with an updated activity for all rules (see Fig. 1.1).

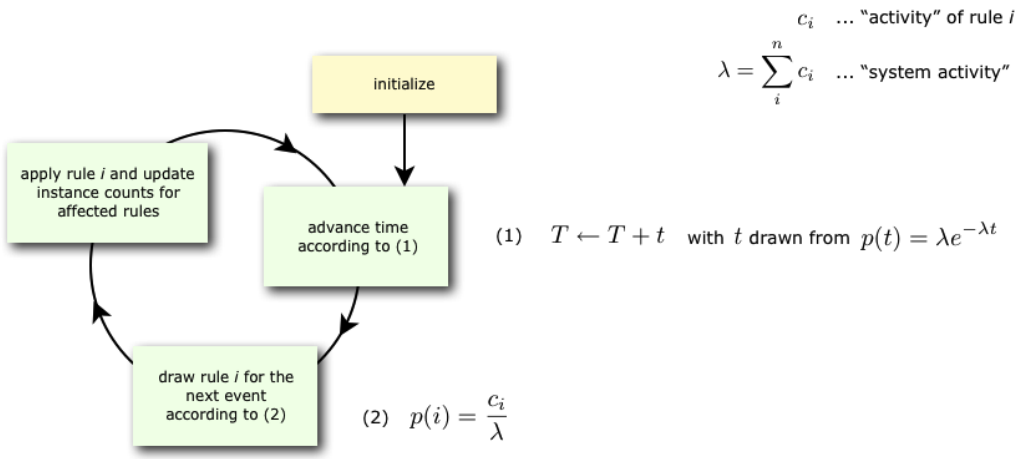

Figure 1.1: The event loop

Importantly, the cost of such an event is independent of the size of the graph it is applied to [1]. Note that KaSim is not equipped with a curve visualization tool. However, the outputted data are in text format and usable with any standard plotting software such as GnuPlot.

Chapter 2

Installation

To obtain KaSim you can either use pre-compiled binaries available on KappaLanguage.org or compile the sources for your architecture. To do so, download the source code from `https://github.com/jkrivine/KaSim` and make sure you have a recent ocaml compiler installed. From a terminal window type `ocamlopt.opt -v`. If nothing appears then you need to install Ocaml Native compiler that can be downloaded from `http://caml.inria.fr/download.en.html`.

Once Ocaml is safely installed, untar KaSim archive and compile following these few steps:

```
$ tar xzvf kasim.tar.gz -d Kappa

$ cd Kappa

$ make
```

At the end of these steps you should see, in the `Kappa` directory, an executable file name KaSim. In order to check the compilation went fine, simply type .\`KaSim --version`. If the ocaml native compiler `ocamlopt.opt` is not in the path of your system, you may set the variable `OCAMLBINPATH` to point to the location of the compiler by editing the corresponding line in the Makefile.

Chapter 3

The command line

Chapter 4

The kappa file

Bibliography

[1] Vincent Danos, Jérome Féret, Walter Fontana, and Jean Krivine. Scalable simulation of cellular signaling networks. In *Proceedings of APLAS'07: 5^th ASIAN symposium on programming languages and systems*, volume 4807 of *LNCS*, pages 139–157, 2007. Invited paper.

[2] Vincent Danos, Jérôme Feret, Walter Fontanta, Russ Harmer, and Jean Krivine. Rule based modeling of biological signaling. In Luís Caires and Vasco Thudichum Vasconcelos, editors, *Proceedings of CONCUR 2007*, volume 4703 of *LNCS*, pages 17–41. Springer, 2007.

[3] Vincent Danos and Cosimo Laneve. Formal molecular biology. *Theoretical Computer Science*, 325, 2004.

[4] James R. Faeder, Mickael L. Blinov, and William S. Hlavacek. Rule based modeling of biochemical networks. *Complexity*, pages 22–41, 2005.

[5] Daniel T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403–434, 1976.

[6] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81(25):2340–2361, 1977.